AMENDMENTS TO THE CLAIMS

Claims 1-25 (Canceled)

26. (Currently Amended) A digital circuit configured as ~~An~~ an arithmetic pipeline, said circuit comprising:

a flat four-input single precision floating point adder module, said module being controllable to add first, second, third and fourth single precision floating point numbers and to output a resulting single precision floating point number, said module comprising:

means for predicting a largest number from exponent and mantissa portions of said floating point numbers, said predicting means outputting a plurality of shifting data calculated based on said largest number and said exponent portions;

means for partially sorting said floating point numbers based on sign-bit and the exponent portions of said floating point numbers, said sorting means outputting sorted mantissas, sorted exponents, and sorted sign-bits;

carry-in generation means for outputting carry-in data based on said sorted sign-bits and mantissas;

addition logic receiving the carry-in data and said sorted mantissas and said plurality of shifting data, said addition logic calculating and outputting a normalized mantissa output and exponent modifier; and

output logic receiving the normalized mantissa output, exponent modifier, and a largest exponent, said output logic outputting the resulting floating point number based on the normalized mantissa output, the exponent modifier, and the largest exponent.

27. (Currently Amended) The digital circuit ~~arithmetic pipeline~~ of claim 26, wherein all arithmetic negations are approximated to a logical negation and said carry-in generation means generates the carry-in data to correct said approximations.

28. (Currently Amended) The digital circuit ~~arithmetic pipeline~~ of claim 26, wherein said carry-in generation means generates the carry-in data to correct any loss of precision that may have occurred in shifting of non-largest mantissas by said addition logic.

29. (Currently Amended) The digital circuit ~~arithmetic pipeline~~ of claim 26, wherein said carry-in generation means generates the carry-in data to correct incorrect determinations of which floating point number is larger.

30. (Currently Amended) The digital circuit ~~arithmetic pipeline~~ of claim 26, wherein said carry-in generation means generates the carry-in data to correctly round the resulting single precision floating point number to meet rounding mode requirements.

31. (Currently Amended) The digital circuit ~~arithmetic pipeline~~ of claim 26 further comprising a floating point multiplier module, said multiplier module inputs the input data and performs a multiply operation and said four-input single precision floating point adder module performs a normalization operation on a result of the multiply operation.

Claims 32-34 (Canceled)

35. (New) A flat four-input single precision floating point adder module, said module being controllable to add first, second, third and fourth single precision floating point numbers and to output a resulting single precision floating point number, said module comprising:

means for predicting a largest number from exponent and mantissa portions of said floating point numbers, said predicting means outputting a plurality of shifting data calculated based on said largest number and said exponent portions;

means for partially sorting said floating point numbers based on sign-bit and the exponent portions of said floating point numbers, said sorting means outputting sorted mantissas, sorted exponents, and sorted sign-bits;

carry-in generation means for outputting carry-in data based on said sorted sign-bits and mantissas;

addition logic receiving the carry-in data and said sorted mantissas and said plurality of shifting data, said addition logic calculating and outputting a normalized mantissa output and exponent modifier; and

output logic receiving the normalized mantissa output, exponent modifier, and a largest exponent, said output logic outputting the resulting floating point number based on the normalized mantissa output, the exponent modifier, and the largest exponent.

36. (New) The module of claim 35, wherein all arithmetic negations are approximated to a logical negation and said carry-in generation means generates the carry-in data to correct said approximations.

37. (New) The module of claim 35, wherein said carry-in generation means generates the carry-in data to correct any loss of precision that may have occurred in shifting of non-largest mantissas by said addition logic.

38. (New) The module of claim 35, wherein said carry-in generation means generates the carry-in data to correct incorrect determinations of which floating point number is larger.

39. (New) The module of claim 35, wherein said carry-in generation means generates the carry-in data to correctly round the resulting single precision floating point number to meet rounding mode requirements.

40. (New) The module of claim 35, further comprising a floating point multiplier module, said multiplier module inputs the input data and performs a multiply operation and said four-input single precision floating point adder module performs a normalization operation on a result of the multiply operation.

41. (New) A processor executing arithmetic operations on vertex data comprising a flat four-input single precision floating point adder module, said module being controllable to add first, second, third and fourth single precision floating point numbers and to output a resulting single precision floating point number, said module comprising:

means for predicting a largest number from exponent and mantissa portions of said floating point numbers, said predicting means outputting a plurality of shifting data calculated based on said largest number and said exponent portions;

means for partially sorting said floating point numbers based on sign-bit and the exponent portions of said floating point numbers, said sorting means outputting sorted mantissas, sorted exponents, and sorted sign-bits;

carry-in generation means for outputting carry-in data based on said sorted sign-bits and mantissas;

addition logic receiving the carry-in data and said sorted mantissas and said plurality of shifting data, said addition logic calculating and outputting a normalized mantissa output and exponent modifier; and

output logic receiving the normalized mantissa output, exponent modifier, and a largest exponent, said output logic outputting the resulting floating point number based on the normalized mantissa output, the exponent modifier, and the largest exponent.

42. (New) The processor executing arithmetic operations on vertex data of claim 41, wherein all arithmetic negations are approximated to a logical negation and said carry-in generation means generates the carry-in data to correct said approximations.

43. (New) The processor executing arithmetic operations on vertex data of claim 41, wherein said carry-in generation means generates the carry-in data to correct any loss of precision that may have occurred in shifting of non-largest mantissas by said addition logic.

44. (New) The processor executing arithmetic operations on vertex data of claim 41, wherein said carry-in generation means generates the carry-in data to correct incorrect determinations of which floating point number is larger.

45. (New) The processor executing arithmetic operations on vertex data of claim 41, wherein said carry-in generation means generates the carry-in data to correctly round the resulting single precision floating point number to meet rounding mode requirements.

46. (New) The processor executing arithmetic operations on vertex data of claim 41, further comprising a floating point multiplier module, said multiplier module inputs the input data and performs a multiply operation and said four-input single precision floating point adder module performs a normalization operation on a result of the multiply operation.